**Index.html**

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Phaser - final game </title>

<script type="text/javascript" src="js/phaser.min.js"></script>

<script type="text/javascript" src="js/process.js"></script>

<script type="text/javascript" src="js/boot.js"></script>

<script type="text/javascript" src="js/preload.js"></script>

<script type="text/javascript" src="js/menu.js"></script>

<script type="text/javascript" src="js/play.js"></script>

<script type="text/javascript" src="js/game.js"></script>

<style type="text/css">

body {

margin: 0;

}

</style>

</head>

<body>

</body>

</html>

**Game.js**

var w = 800,h = 600;

var player,enemy, btnrestart, menu, destroy, keyboard, circle, circles, score = 0, scoreText, gameOverText, bestScoreText, stateText, introText, lifeText, life = 3, time, timeText, bestText, startButton,startGame, diamond, diamonds;

var btn\_pause ,playing;

var player;

var menuText, playText, aboutText;

var diamond, circle, sky, btnUp, btnDown, btnLeft, btnRight, startButton, player, enemy, enemy1, enemy2, enemy3;

var game = new Phaser.Game(w, h, Phaser.CANVAS, '');

game.state.add("bootGame",bootGame);

game.state.add("preloadGame",preloadGame);

game.state.add("menuGame",menuGame);

game.state.add("playGame",playGame);

game.state.start("bootGame");

**Preload.js**

preloadGame = {

preload:function () {

game.load.image('diamond','img/diamond.png');

game.load.image('circle','img/circle.png');

game.load.image('sky','img/sky.png');

game.load.image('btnUp', 'img/btnUp.png');

game.load.image('btnDown', 'img/btnDown.png');

game.load.image('btnLeft', 'img/btnLeft.png');

game.load.image("buttonplay","img/playButton.png");

game.load.image('btnRight', 'img/btnRight.png');

game.load.image('ins','img/instruction.png');

game.load.image('intruct',"img/patakaran.png");

game.load.image('About','img/aboutbtn.png');

game.load.image('about2','img/about2.png');

//game.load.spritesheet("menu2","img/menu2.png",95,94);

game.load.spritesheet('dude','img/dude.png',32,48);

game.load.spritesheet("menu2","img/menu2.png",80,50);

game.load.spritesheet('startButton','img/start.png',800,600);

game.load.spritesheet('btn-play','img/btn-green.png',285,250);

game.load.spritesheet('player', 'img/characters.png', 16, 16);

game.load.spritesheet('enemy', 'img/characters.png', 16, 16);

game.load.spritesheet('enemy1', 'img/characters.png', 16, 16);

game.load.spritesheet('enemy2', 'img/characters.png', 16, 16);

game.load.spritesheet('enemy3', 'img/characters.png', 16, 16);

//game.load.audio('bg', 'audio/bgmusic.mp3');

},

create:function(){

game.state.start("menuGame");

},

}

**Boot.js**

bootGame = {

create:function () {

game.physics.startSystem(Phaser.Physics.ARCADE);

game.scale.scaleMode = Phaser.ScaleManager.SHOW\_ALL;

game.scale.forceLandscape = true;

game.scale.pageAlignHorizontally = true;

keyboard = game.input.keyboard.createCursorKeys();

game.state.start("preloadGame");

}

}

**Menu.js**

menuGame = {

create:function () {

//game.physics.startSystem(Phaser.Physics.ARCADE);

game.add.sprite(0,0,"sky");

startButton = game.add.button(290,150, "buttonplay",this.buttonPlay);

aboutText = game.add.button(440,530,"About",this.about);

aboutText.anchor.set(0.6);

aboutText.scale.set(1);

menuText = game.add.text(350, 50,"Menu",{"fill":"#64a2f"});

menuText.scale.x = 2;

menuText.scale.y = 2;

instruc = game.add.button(440,380,"intruct",this.ins);

instruc.anchor.set(0.6);

instruc.scale.set(1);

},

about: function(){

about=game.add.image(0,0,"about2");

//about.scale.set(3.2);

restartButton=game.add.button(30,30,"menu2",restartB,this);

function restartB() {

restartButton.destroy();

game.state.start("menuGame");

}

},

ins: function(){

about=game.add.image(0,0,"ins");

//about.scale.set(3.2);

restartButton=game.add.button(30,30,"menu2",restartB,this);

function restartB() {

restartButton.destroy();

game.state.start("menuGame");

}

},

buttonPlay:function(){

game.state.start("playGame");

//menumusic.stop();

},

update:function(){

//if(keyboard.up.isDown){

// game.state.start("playGame");

//}

}

}

**Process.js**

var proseso = function(){

"use strict";

return {

enemyMoveDown:function() {

enemy1.body.velocity.y = 500;

enemy1.animations.play('down');

enemy.body.velocity.y = 500;

enemy.animations.play('down');

enemy2.body.velocity.y = 500;

enemy2.animations.play('down');

enemy3.body.velocity.y = 500;

enemy3.animations.play('down');

},

enemyMoveUp:function() {

enemy1.body.velocity.y =-500;

enemy1.animations.play('up');

enemy3.body.velocity.y =-500;

enemy3.animations.play('up');

},

enemyMoveRight:function() {

enemy.body.velocity.x = 500;

enemy.animations.play('right');

},

enemyMoveLeft:function() {

enemy.body.velocity.x = -500;

enemy.animations.play('left');

},

enemy2MoveLeft:function() {

enemy2.body.velocity.x = -500;

enemy2.animations.play('left');

},

enemy2MoveRight:function() {

enemy2.body.velocity.x = 500;

enemy2.animations.play('right');

},

enemy3MoveRight:function() {

},

enemy3MoveLeft:function() {

},

collectDiamonds:function (player, diamond){

//var score = 0;

score = score + 5;

diamond.kill();

if(proseso.getScore()<=score){

saveScore(a);

bestScoreText.text = "Best: "+score;

}

time.text = "Score: "+score;

},

killCharacters:function (player,enemy, enemy2, enemy3){

game.\_paused = true;

stateText.text =" GAME OVER\n\n Tap to Restart";

stateText.visible = true;

game.input.onTap.addOnce(proseso.restart,this);

},

saveScore:function (Score){

localStorage.setItem("gameScore",Score);

},

getScore:function (){

return (localStorage.getItem("gameScore") == null || localStorage.getItem("gameScore") == "")?0:localStorage.getItem("gameScore");

},

pushup:function (){

player.animations.play('walk-up');

player.body.velocity.y=-100;

},

pushdown:function (){

player.animations.play('walk-down');

player.body.velocity.y=100;

},

pushright:function (){

player.animations.play('walk-right');

player.body.velocity.x=100;

},

pushleft:function (){

player.animations.play('walk-left');

player.body.velocity.x=-100;

},

startGame:function (){

startButton.destroy();

player.body.velocity.set(0,0);

proseso.laying = true;

},

playGame:function (){

setTimeout(function(){

button\_Galaw.frame = 0;

game.\_paused = false;

},5000);

game.\_paused = true;

button\_Galaw.frame = 1;

},

restart:function () {

window.location.href=window.location.href;

stateText.visible = false;

}

}

}();

**Play.js**

playGame = {

create:function () {

game.physics.startSystem(Phaser.Physics.ARCADE);

game.scale.scaleMode = Phaser.ScaleManager.SHOW\_ALL;

game.scale.forceLandscape = true;

game.scale.pageAlignHorizontally = true;

game.add.sprite(0,0,"sky");

btnRight = game.add.button(80,400,"btnUp",proseso.pushup);

btnLeft = game.add.button(80,500,"btnDown",proseso.pushdown);

btnRight = game.add.button(700,500,"btnRight",proseso.pushright);

btnLeft = game.add.button(600,500,"btnLeft",proseso.pushleft);

circle = game.add.group();

circle.enableBody = true;

circles = circle.create(150, 80,"circle");

circles.body.immovable = true;

diamond = game.add.group();

diamond.enableBody = true;

diamonds = diamond.create(350, 290,"diamond");

diamonds = diamond.create(250, 320,"diamond");

diamonds = diamond.create(350, 180,"diamond");

diamonds = diamond.create(390, 400,"diamond");

diamonds = diamond.create(550, 460,"diamond");

diamonds = diamond.create(230, 230,"diamond");

diamonds = diamond.create(420, 230,"diamond");

diamonds = diamond.create(450, 280,"diamond");

diamonds = diamond.create(510, 380,"diamond");

diamonds = diamond.create(550, 310,"diamond");

diamonds = diamond.create(310, 360,"diamond");

diamonds = diamond.create(580, 340,"diamond");

diamonds = diamond.create(300, 210,"diamond");

diamonds = diamond.create(420, 130,"diamond");

diamonds = diamond.create(560, 230,"diamond");

diamonds = diamond.create(260, 430,"diamond");

diamonds = diamond.create(460, 460,"diamond");

diamonds = diamond.create(360, 460,"diamond");

time = game.add.text(620,40,"SCORE: 0",{fill:'yellow'});

bestScoreText = game.add.text(620,100,'Best: '+proseso.getScore(),{fill:"yellow"});

stateText = game.add.text(300,200,' ', { font: '30px times new roman', fill: 'blue' });

//stateText.anchor.setTo(0.5, 0.5);

stateText.visible = false;

button\_Galaw = game.add.button(50,0,"btn-play",proseso.playGame);

button\_Galaw.scale.x=0.5;

button\_Galaw.scale.y=0.5;

// bgmusic = game.add.audio('bgmusic');

// bgmusic.play().loopFull();

player = game.add.sprite(288,48,'dude');

keyboard = game.input.keyboard.createCursorKeys();

player.scale.x = 1.3;

player.scale.y = 1.3;

player.animations.add('walk-left',[0,1,2,3],6,true);

player.animations.add('walk-right',[6,7,8,9],6,true);

player.animations.stop();

game.physics.arcade.enable(player);

player.body.collideWorldBounds = true;

enemy = game.add.sprite(400,270,"enemy");

enemy.animations.add('left', [21, 22, 23], 10, true);

enemy.animations.add('right', [33, 34, 35], 10, true);

enemy.scale.x=3;

enemy.scale.y=3;

game.physics.arcade.enable(enemy);

enemy.body.collideWorldBounds = true;

enemy1 = game.add.sprite(330,240,"enemy1");

enemy1.animations.add('down', [9, 10, 11], 10, true);

enemy1.animations.add('up', [45, 46, 47], 10, true);

enemy1.scale.x=3;

enemy1.scale.y=3;

game.physics.arcade.enable(enemy1);

enemy1.body.collideWorldBounds = true;

enemy2 = game.add.sprite(350,150,"enemy2");

enemy2.animations.add('left', [21, 22, 23], 10, true);

enemy2.animations.add('right', [33, 34, 35], 10, true);

enemy2.scale.x=3;

enemy2.scale.y=3;

game.physics.arcade.enable(enemy2);

enemy2.body.collideWorldBounds = true;

enemy3 = game.add.sprite(420,310,"enemy3");

enemy3.animations.add('down', [9, 10, 11], 10, true);

enemy3.animations.add('up', [45, 46, 46], 10, true);

enemy3.scale.x=3;

enemy3.scale.y=3;

game.physics.arcade.enable(enemy3);

enemy3.body.collideWorldBounds = true;

timer=game.time.events.loop(Phaser.Timer.SECOND \* 0.6,proseso.enemyMoveDown);

timer=game.time.events.loop(Phaser.Timer.SECOND \* 1,proseso.enemyMoveRight);

timer=game.time.events.loop(Phaser.Timer.SECOND \* 0.9,proseso.enemy2MoveRight);

timer=game.time.events.loop(Phaser.Timer.SECOND \* 0.7,proseso.enemy3MoveRight);

timer=game.time.events.loop(Phaser.Timer.SECOND \* 1,proseso.enemyMoveUp);

timer=game.time.events.loop(Phaser.Timer.SECOND \* 2,proseso.enemyMoveLeft);

timer=game.time.events.loop(Phaser.Timer.SECOND \* 1.5,proseso.enemy2MoveLeft);

timer=game.time.events.loop(Phaser.Timer.SECOND \* 1.6,proseso.enemy3MoveLeft);

startButton= game.add.button(400,300,'startButton',proseso.startGame);

startButton.anchor.set(0.5);

game.\_puased= true;

},

update:function () {

game.physics.arcade.collide(enemy,diamonds);

game.physics.arcade.collide(enemy2,diamonds);

game.physics.arcade.collide(enemy3,diamonds);

game.physics.arcade.overlap(player, diamond,proseso. collectDiamonds);

game.physics.arcade.overlap(player,enemy,proseso.killCharacters);

game.physics.arcade.overlap(player,enemy1,proseso.killCharacters);

game.physics.arcade.overlap(player,enemy2,proseso.killCharacters);

game.physics.arcade.overlap(player,enemy3,proseso.killCharacters);

function pushup()

{

player.body.velocity.y=100;

player.animations.play('walk-up');

bg.tilePosition.x -=10;

}

function pushdown()

{

player.animations.play('walk-down');

player.body.velocity.y=-200;

//player.animations.add("walk-left",[1,2,3,4],5,true);

bg.tilePosition.x +=-10;

}

function pushright()

{

player.body.velocity.x=-100;

player.animations.play('walk-right');

bg.tilePosition.x -=10;

}

function pushleft()

{

player.animations.play('walk-left');

player.body.velocity.x=-200;

//player.animations.add("walk-left",[1,2,3,4],5,true);

bg.tilePosition.x +=-10;

}

},

enemyMoveDown:function() {

enemy1.body.velocity.y = 500;

enemy1.animations.play('down');

enemy.body.velocity.y = 500;

enemy.animations.play('down');

enemy2.body.velocity.y = 500;

enemy2.animations.play('down');

enemy3.body.velocity.y = 500;

enemy3.animations.play('down');

},

enemyMoveUp:function() {

enemy1.body.velocity.y =-500;

enemy1.animations.play('up');

// enemy.body.velocity.y =-500;

// enemy.animations.play('up');

// enemy2.body.velocity.y =-500;

// enemy2.animations.play('up');

enemy3.body.velocity.y =-500;

enemy3.animations.play('up');

},

enemyMoveRight:function() {

enemy.body.velocity.x = 500;

enemy.animations.play('right');

},

enemyMoveLeft:function() {

enemy.body.velocity.x = -500;

enemy.animations.play('left');

},

enemy2MoveLeft:function() {

enemy2.body.velocity.x = -500;

enemy2.animations.play('left');

},

enemy2MoveRight:function() {

enemy2.body.velocity.x = 500;

enemy2.animations.play('right');

},

enemy3MoveRight:function() {

//enemy3.body.velocity.x = 500;

//enemy3.animations.play('right');

},

enemy3MoveLeft:function() {

//enemy3.body.velocity.x = -500;

//enemy3.animations.play('left');

},

//var a = 0;

collectDiamonds:function (player, diamond){

//var score = 0;

score = score + 5;

diamond.kill();

if(proseso.getScore()<=score){

saveScore(a);

bestScoreText.text = "Best: "+score;

}

time.text = "Score: "+score;

},

killCharacters:function (player,enemy, enemy2, enemy3){

game.\_paused = true;

//player.kill();

stateText.text =" GAME OVER\n\n Tap to Restart";

stateText.visible = true;

game.input.onTap.addOnce(proseso.restart,this);

},

saveScore:function (Score){

localStorage.setItem("gameScore",Score);

},

getScore:function (){

return (localStorage.getItem("gameScore") == null || localStorage.getItem("gameScore") == "")?0:localStorage.getItem("gameScore");

},

pushup:function (){

player.animations.play('walk-up');

player.body.velocity.y=-100;

//bg.tilePosition.x -=4;

},

pushdown:function (){

player.animations.play('walk-down');

player.body.velocity.y=100;

//bg.tilePosition.x -=4;

},

pushright:function (){

player.animations.play('walk-right');

player.body.velocity.x=100;

//bg.tilePosition.x -=4;

},

pushleft:function (){

player.animations.play('walk-left');

player.body.velocity.x=-100;

//bg.tilePosition.x -=4;

},

startGame:function (){

startButton.destroy();

player.body.velocity.set(0,0);

proseso.laying = true;

},

playGame:function (){

setTimeout(function(){

button\_Galaw.frame = 0;

game.\_paused = false;

},5000);

game.\_paused = true;

button\_Galaw.frame = 1;

},

restart:function () {

window.location.href=window.location.href;

stateText.visible = false;

}

}